Getting Started – DK241
IPC@CHIP® Embedded Web Controller Family

Development Kit DK241 Order No. 563965

Copyright © 2009 Beck IPC GmbH

Table of Contents

1 INTRODUCTION 3
2 SYSTEM START-UP 3
2.1 WHAT DO WE NEED TO GET STARTED? 3
2.2 STEP 1 – SWITCHING ON THE DB240 DEVELOPMENT BOARD 4
2.3 STEP 2 – STARTING THE ONE-WORKBENCH 5
2.4 STEP 3 – SETTING UP COMMUNICATION 7
2.4.1 Setting up the serial communication 7
2.4.2 Setting up the Ethernet communication 8
2.5 STEP 4 – DOWNLOADING THE DB240 ONLINE DEMO PROGRAM 14
2.6 STEP 5 – OPERATING THE DEMO PROGRAM 16
2.7 STEP 6 – CONFIGURING SD/MMC-CARD 18
3 INFORMATION FOR ADVANCED USERS 19
3.1 UPDATING THE @CHIP-RTOS-PPC 19
3.2 UPDATING THE ONE-WORKBENCH 20
3.2.1 Show currently installed features 20
3.2.2 Find and install updates and new features 21
3.3 USING THE IPC@CHIPTOOL APPLICATION 24
3.4 DEVELOPING SOFTWARE FOR YOUR IPC@CHIP® 25
3.5 DEBUGGING YOUR IPC@CHIP® APPLICATIONS 28
3.6 MISCELLANEOUS 31
3.6.1 The CHIP.INI configuration file 31
3.6.2 The command shell 31
3.6.3 The web server 31
3.6.4 The PPP server/client 32
3.6.5 IPv4 Basics 33
3.6.6 IPv4/v6 Dual Stack 34
4 APPENDIX - WHERE TO GET FURTHER INFORMATION 35
4.1 GETTING STARTED 35
4.2 UPDATES, TOOLS AND EXAMPLES 35
4.3 SUPPORT 35
4.4 ORDER NUMBERS 35

Copyright © 2009 Beck IPC GmbH

1 Introduction

Congratulations on purchasing your IPC@CHIP® DK241 Development Kit. The Development Kit is a complete ready-to-use development system for building applications based on the Embedded Web Controller IPC@CHIP® SC243. It runs with the preinstalled @CHIP-RTOS-PPC operating system, which includes features like a real-time kernel that is capable of running applications concurrently, a full TCP/IP stack with a UDP/TCP socket interface, DHCP, FTP, PPP, Telnet, SSH server, Web server, APIs for CAN and USB controllers, etc.
The Development Kit contains both hardware and software components required for a basic development. This document will introduce you to the several components of the DK241. Chapter 2 provides a step-by-step guide on how to put the DK241 into operation. Chapter 3 describes selected components of the DK241 for advanced users. Chapter 4 points the way to further information.

2 System Start-up

2.1 What do we need to get started?

- The Development Kit DK241 consisting of:
  - Development Board DB240
  - Embedded Web Controller IPC@CHIP® SC243 (on the DB240)
  - Serial null-modem cable
  - Ethernet patch cable (gray)
  - Ethernet cross-over cable (red)
  - Secure Digital (SD) memory card
  - Power supply unit
  - ONE-Workbench for IPC@CHIP® CD (including GNU C/C++ compiler, DK241 startup documentation and drivers)
- Personal Computer
  - Microsoft Windows XP/Vista Operating System
  - Ethernet network interface
  - Serial port

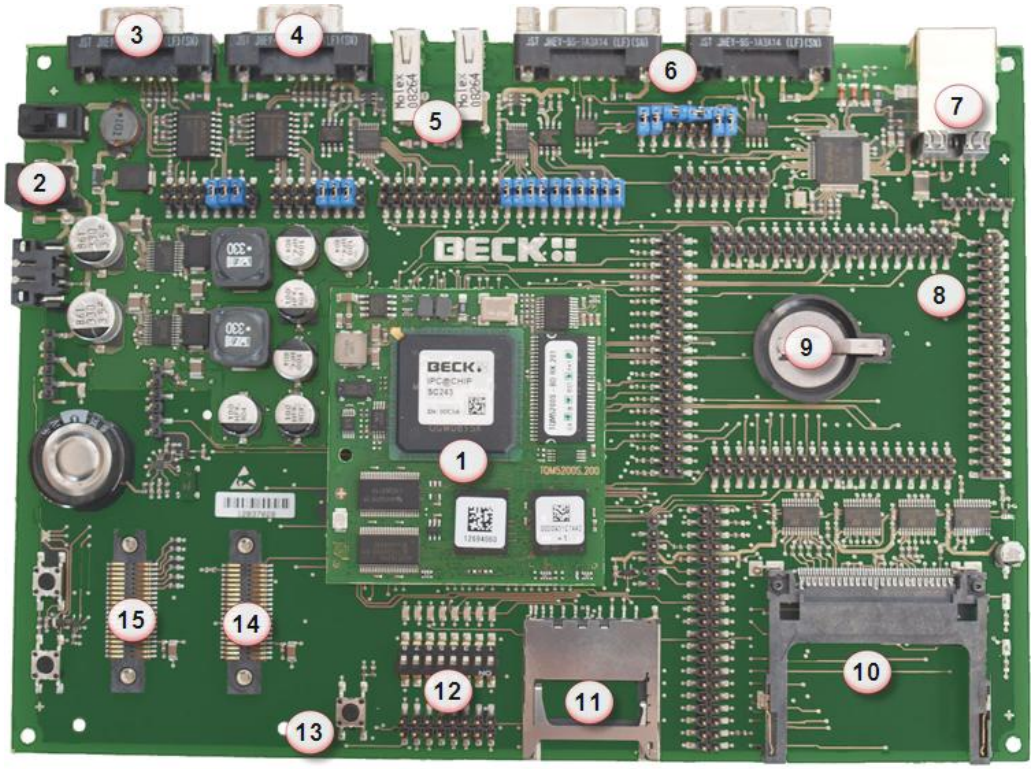
Software from the enclosed ONE-Workbench CD or the IPC@CHIP® Getting Started Website at: http://www.beck-ipc.com/gettingstarted
- ONE-Workbench (on CD only)
- DB240 Visualisation Demo
- EXTDISK driver software
- @CHIPTOOL

Be aware that the Embedded Controller SC243 and the Development Board DB240 are electrostatic sensitive components. Observe general precautions for handling.

Copyright © 2009 Beck IPC GmbH

2.2 Step 1 – Switching on the DB240 Development Board

The Development board DB240 is designed to give you easy access to the functionality of the IPC@CHIP® SC243, especially to its several interfaces like serial ports, Ethernet, digital I/O, CAN, USB, etc. The following graphic gives an overview of the most important components of the DB240 board.



1	SC243	8	Extension socket
2	Power supply	9	Battery for SRAM
3	UART1 (Shell & Bootloader)	10	CF card socket
4	UART3	11	MMC / SD-card socket
5	2 x USB	12	LEDs and Dip-Switches
6	2 x CAN	13	Reset switch
7	Ethernet	14	WLAN socket
		15	Bluetooth socket

The AC power supply is delivered with country specific AC adapters. Please choose the right adapter for your country. Then connect the AC power supply to your main supply and to the DB240 Development Board to power it. Move the power switch on the DB240 to the *ON* position, two green Power LEDs will light up.

Copyright © 2009 Beck IPC GmbH

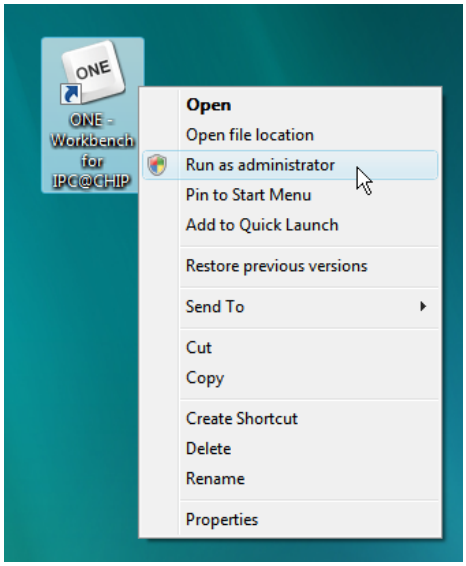
2.3 Step 2 – Starting the ONE-Workbench

The ONE-Workbench for IPC@CHIP® is a comfortable integrated development environment for the SC243/SC223 products. It integrates the GNU C/C++ toolchain for application development and debugging. It also provides functionality for finding IPC@CHIP®s on the network, configuring them, updating their operation system, transferring files and using their shell. The workbench is based on the common Eclipse framework and the CDT C/C++ editor plug-in. It benefits from a large user community which results in plenty of features for modern and comfortable software development.

For installing the ONE-Workbench at your PC, execute the setup program `ONE-WORKBENCH-x.x.x-INSTALLER.EXE` from the CD.

It's strongly discouraged to install the ONE-Workbench to the Windows program files directory ("`C:\Program Files\`"), because this may lead to problems during the first program launch and during the online update, especially if you run with only normal user permissions. It's recommended to install it in some other location, e.g. "`C:\ONE-Workbench\`".

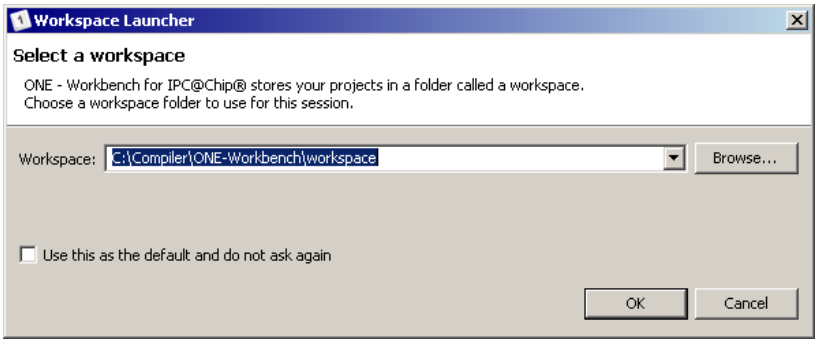
Note: If you anyhow install the ONE-Workbench to the Windows program files directory, make sure that you have administrator rights during the setup and for the first program launch, because the ONE-Workbench will extract the toolchain on the first launch. Please compile a program one-time with administrator permissions to assure that the toolchain was completely and successfully extracted. See the following screenshot how to start the ONE-Workbench as administrator under Windows Vista, using a right mouse click on the ONE-Workbench desktop icon.



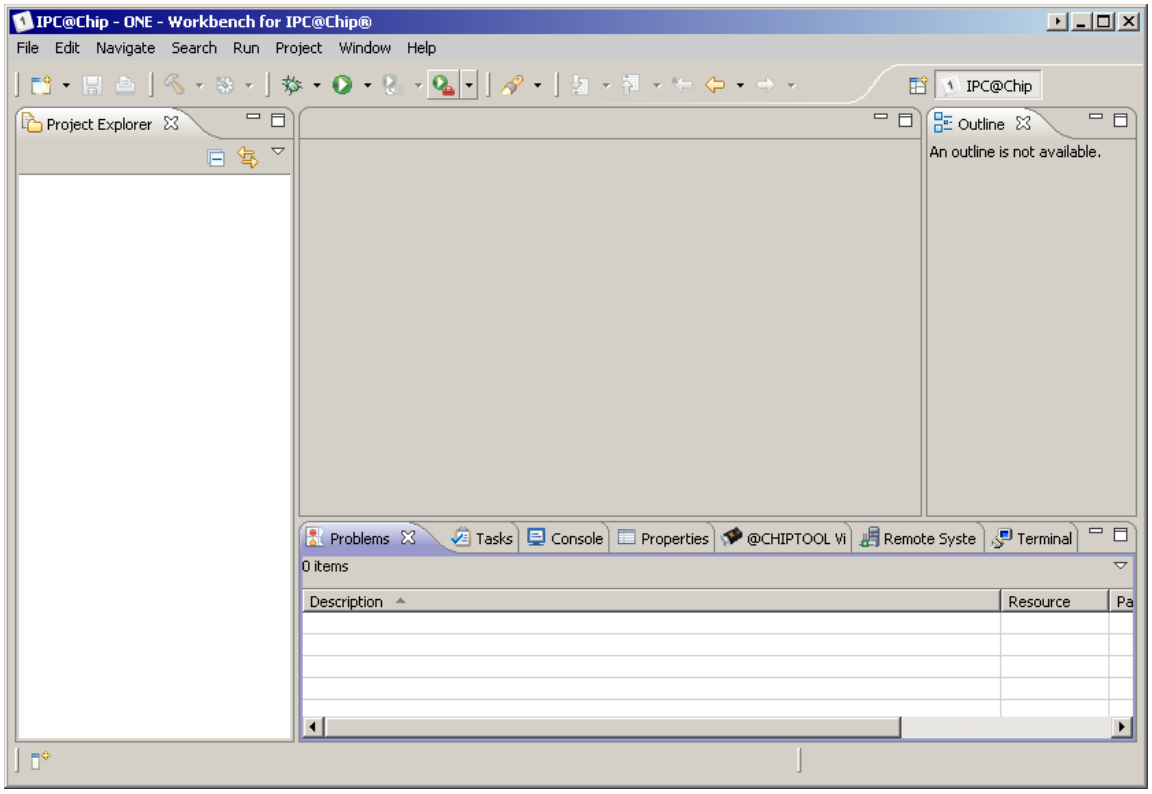
After the installation, you are able to start the ONE-Workbench application from either the Windows start menu or from a desktop icon.

Copyright © 2009 Beck IPC GmbH

During the start of the ONE-Workbench you will be asked to choose an appropriate workspace folder. Inside of this directory the ONE-Workbench will store your project information and preference settings.


Workspace Launcher
Select a workspace
ONE - Workbench for IPC@Chip® stores your projects in a folder called a workspace.
Choose a workspace folder to use for this session.
Workspace:
C:\Compiler\ONE-Workbench\workspace
Browse...
Use this as the default and do not ask again
OK
Cancel

The following graphic shows the main window of the ONE-Workbench.



The ONE-Workbench comes with detailed help. Select *Help Contents* from the *Help* menu for further information.

If you're running a firewall, make sure that it does not block any network functions of the ONE-Workbench such as Telnet, SSH, FTP or the scanning for IPC@CHIP®s.

Copyright © 2009 Beck IPC GmbH

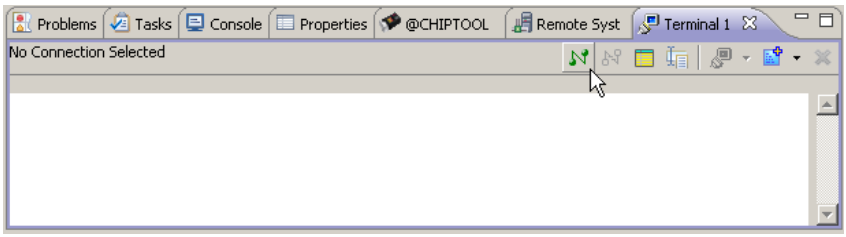
2.4 Step 3 – Setting up communication

There are basically two ways to connect your DB240 to your PC: the serial interface and Ethernet. The simplest connection type is the serial interface. Preferably use it to get access to the shell of your IPC@CHIP®. Ethernet communication is the most powerful. It may be used to access the shell via Telnet or SSH and it's the easiest way to transfer files to the IPC@CHIP® via FTP.

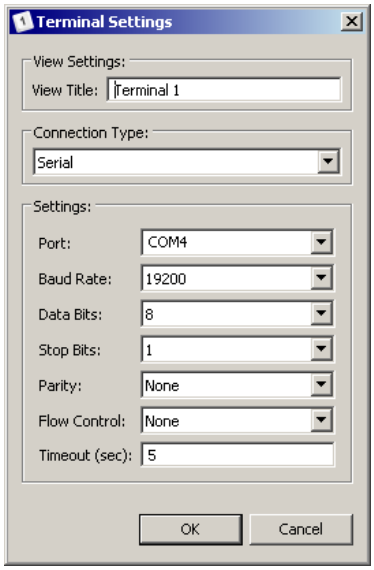
2.4.1 Setting up the serial communication

To set up serial communication to your Development Board DB240 you need the enclosed null-modem cable.

1. Connect the null-modem cable to a free serial port of your PC and with the UART1 interface of the DB240.
2. Select the *Terminal* window in the ONE-Workbench.
3. Click on the green *Connect* button.



4. Select the *Serial* connection type and the PC COM port, where you have connected the null-modem cable. Set the Baud Rate to *19200*, Data Bits to *8*, Stop Bits to *1* and Parity and Flow Control to *None*.

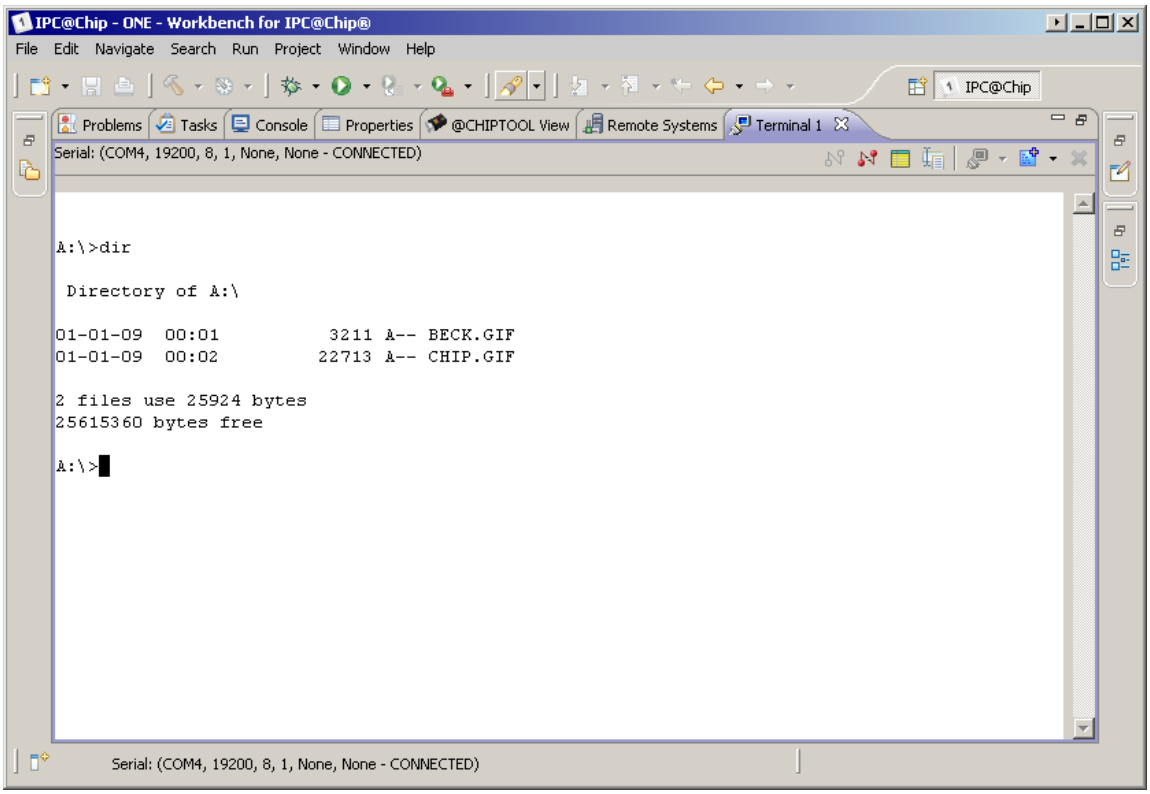


5. Click *OK*. You're now connected to your IPC@CHIP®. Press the Return key in the Terminal window to see the IPC@CHIP® prompt.

Copyright © 2009 Beck IPC GmbH

Hint: By the way, if you double-click on the title line of a ONE-Workbench window, it will be maximized. If you double-click again, it will be restored to its original size.

You may now use the commands of the @CHIP-RTOS-PPC shell in the terminal window. The command `DIR` will, for example, display a list of the files shipped on your IPC@CHIP® internal drive.


IPC@Chip - ONE - Workbench for IPC@Chip®
File Edit Navigate Search Run Project Window Help
IPC@Chip
Problems Tasks Console Properties @CHIPTOOL View Remote Systems Terminal 1
Serial: (COM4, 19200, 8, 1, None, None - CONNECTED)
A:\>dir
Directory of A:\
01-01-09 00:01 3211 A-- BECK.GIF
01-01-09 00:02 22713 A-- CHIP.GIF
2 files use 25924 bytes
25615360 bytes free
A:\>
Serial: (COM4, 19200, 8, 1, None, None - CONNECTED)

2.4.2 Setting up the Ethernet communication

Network communication is based on the TCP/IP protocol family. To be able to operate in a TCP/IP network, your IPC@CHIP® needs to be assigned a valid and unambiguous IP address and network mask. If you cannot figure out a proper IP address and network mask for your IPC@CHIP®, please consult your network administrator.

There are two ways of assigning an IP address to your IPC@CHIP® SC243. You may either configure it by hand or you may use DHCP to automatically assign IP settings. Of course you will need a DHCP server to achieve the latter.

The default configuration for the IPC@CHIP® is using DHCP since this will ensure that the IPC@CHIP® adapts to an existing network if this network provides a DHCP server. If your network doesn't have a DHCP server, the IPC@CHIP® will not disturb the operation of your network.

If you have a regular local network with several clients, connect the DB240's Ethernet interface to your Ethernet hub or switch using the enclosed twisted pair patch cable (gray) and go on to either section 2.4.2.1 or 2.4.2.2, depending on how you'd like to assign IP settings to your IPC@CHIP®.

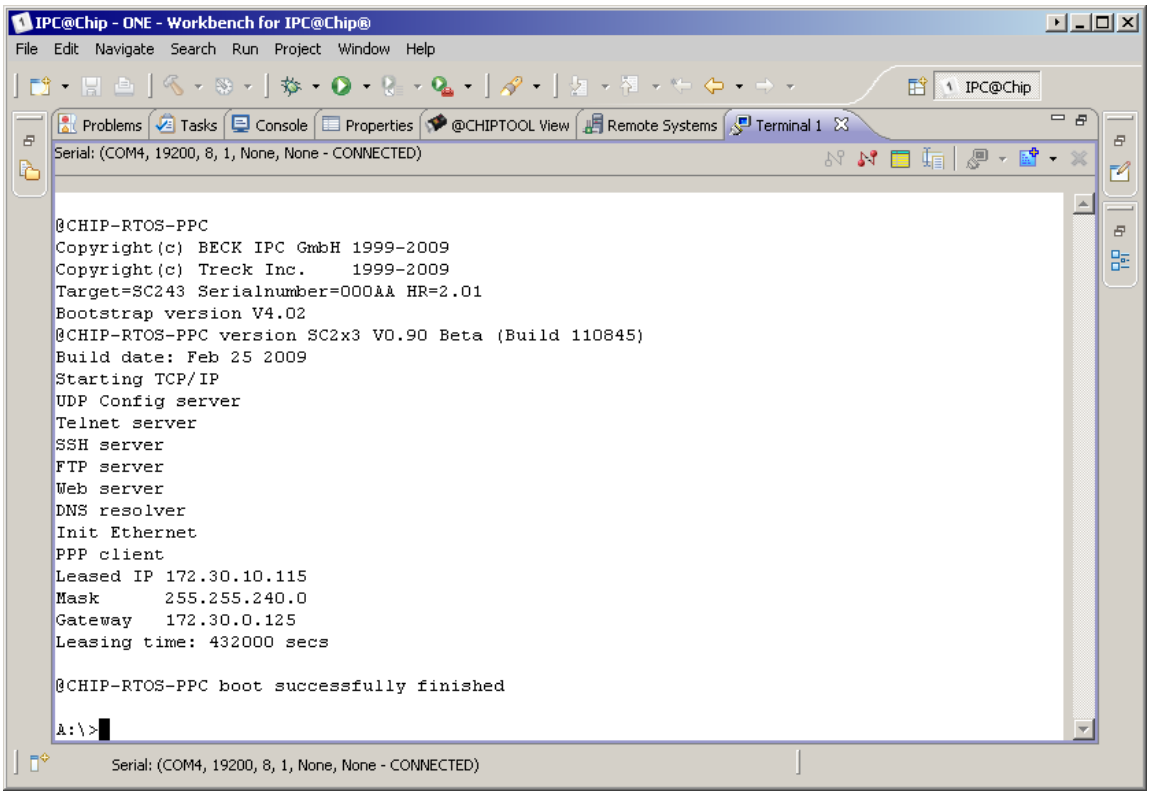
If you're not connected to a network but want to connect the DB240 directly to your PC, you have to use the enclosed cross-over twisted pair Ethernet cable (red) to plug the DB240's Ethernet interface to your PC's Ethernet interface. Then go on to section 2.4.2.2.

Copyright © 2009 Beck IPC GmbH

2.4.2.1 IP address configuration via DHCP

In a network supporting DHCP the IPC@CHIP® retrieves a valid IP configuration from a DHCP server automatically.

Press the Reset button of your DB240. On the serial terminal window the boot messages will show up. Here you will see the serial number, @CHIP-RTOS-PPC version, build date and other information. If an IP configuration could be successfully received from the DHCP server, this IP configuration will be displayed at the end of the boot messages. In the event of an error, the message *DHCP request failed* will be put out after a few seconds.



Copyright © 2009 Beck IPC GmbH

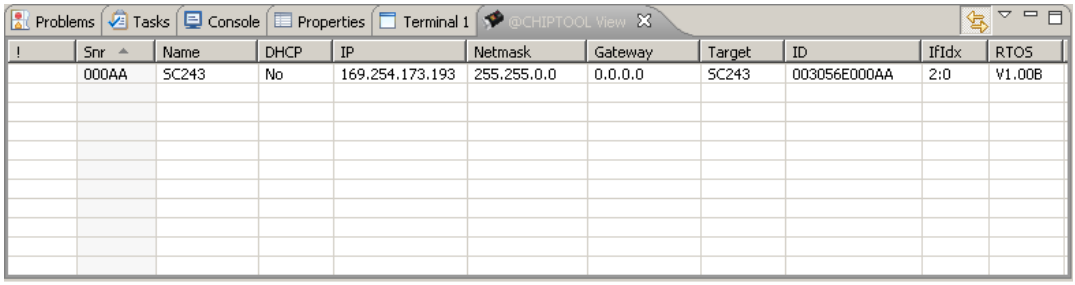
2.4.2.2 IP address configuration by hand

The easiest way to configure your IPC@CHIP®'s IP settings by hand is to use the functionality of the ONE-Workbench *@CHIPTOOL View* window.

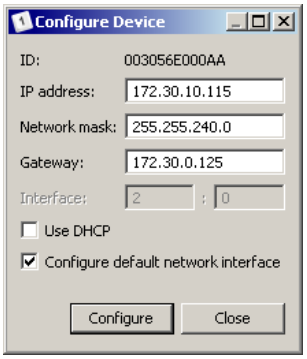
Hint: This window is normally shown at the bottom of the ONE-Workbench application. If it's not shown, select "Show view / @CHIPTOOL view" from the "Window" menu.

In the *@CHIPTOOL View* window you can see all IPC@CHIP®s found on the network. The table displays information such as serial number, name, IP configuration and RTOS version.

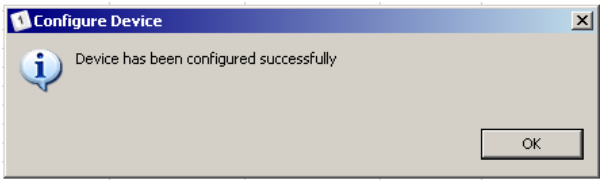
In a network without a DHCP server your IPC@CHIP® should be displayed as *Not configured* or with an IP address from the Auto-IP address range (169.254.x.x), since by default DHCP and AUTOIP is enabled but no DHCP server could be found.



1. Right-click on your IPC@CHIP®.
2. Select *Configure device…* from the pop-up menu.
3. In the appearing dialog uncheck *Use DHCP* and check *Configure default network interface*.
4. Fill the fields *IP Address*, *Network mask* and, if necessary, *Gateway* with appropriate values.
5. Press the *Configure* button to apply the new settings to your IPC@CHIP®.



6. Close the dialog with the *OK* button.



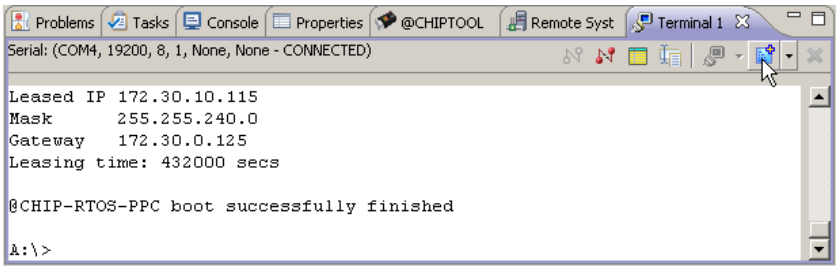
7. Go back to the *@CHIPTOOL View* window. It now shows the new IP configuration.

Copyright © 2009 Beck IPC GmbH

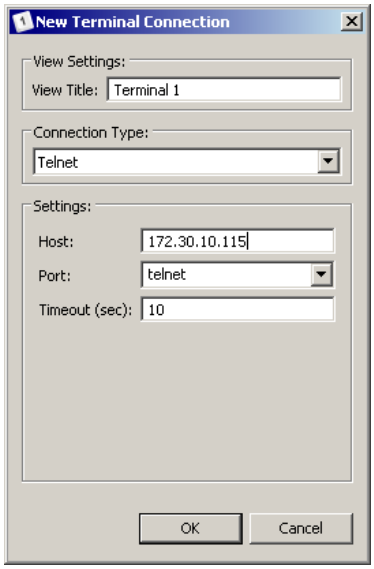
2.4.2.3 Using Telnet to access the shell

Instead of using the serial terminal as described in section 2.4.1 to operate the shell of the @CHIP-RTOS-PPC you may use Telnet. Once the IP configuration of your IPC@CHIP® is completed, simply perform the following steps:

1. In the terminal window select the blue icon to setup a new terminal connection.



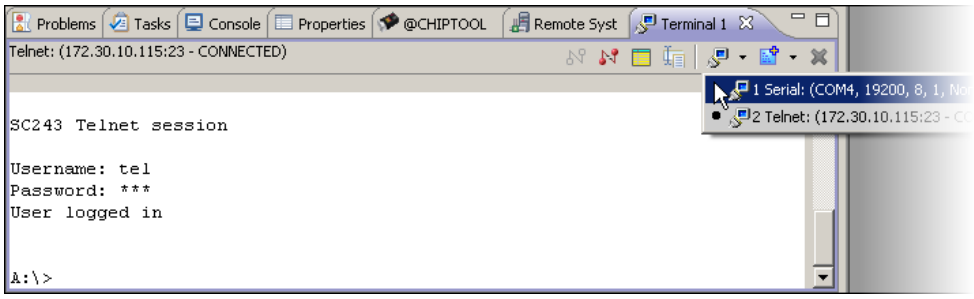
2. Select *Telnet* in the *Connection Type* field and enter your IPC@CHIP®'s IP address in the *Host* field.



3. Press OK. You're now logged in to your IPC@CHIP®.
4. Enter the default user name: *tel*
5. Enter the default password: *tel*
6. Press Return to see the prompt.

You may use telnet just like the serial terminal but the connection will be closed if the IPC@CHIP® restarts.

You can switch between the serial connection and the telnet connection by clicking on the connection symbol in the taskbar.



Copyright © 2009 Beck IPC GmbH

2.4.2.4 Setup a SSH and FTP connection to the IPC@CHIP®

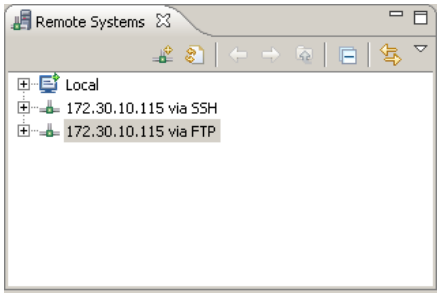
Instead of the Telnet protocol you can also setup a SSH connection to operate the shell of the @CHIP-RTOS-PPC. The FTP protocol can be used to access the file system of the IPC@CHIP®.

Perform the following steps to setup a SSH and FTP connection within the ONE-Workbench.

1. Go to the @CHIPTOOL window.
2. Right-click on your IPC@CHIP® once again.
3. Select *Add Connection* from the pop-up menu.

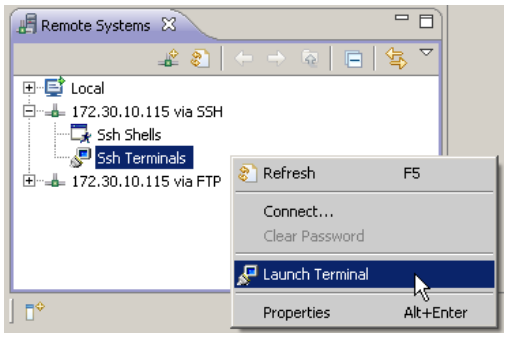
The *Remote Systems* window should now list three trees. If the window is not visible, go to the *Windows / Show View* menu and select the *Remote Systems* entry.

The *Local* tree lists the files on your local PC. Below this tree there should now be two further trees with the IP address of your IPC@CHIP®.

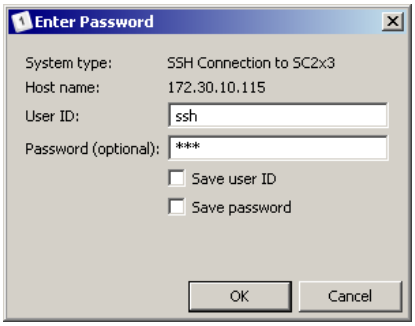


Perform the next steps to establish a SSH shell connection.

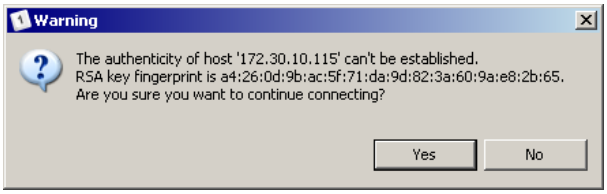
4. Open the tree of the *via SSH* connection and select the *Ssh Terminals* node.
5. Right-click and select the *Launch Terminal* entry from the pop-up menu.



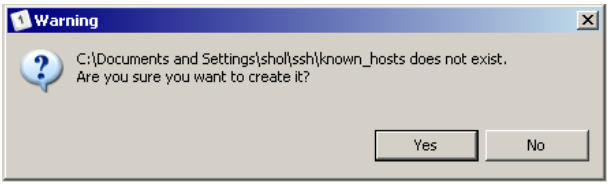
6. Complete the fields *User ID* and *Password* each with *ssh*



7. Accept the fingerprint of the IPC@CHIP® SSH server by selecting *Yes*.

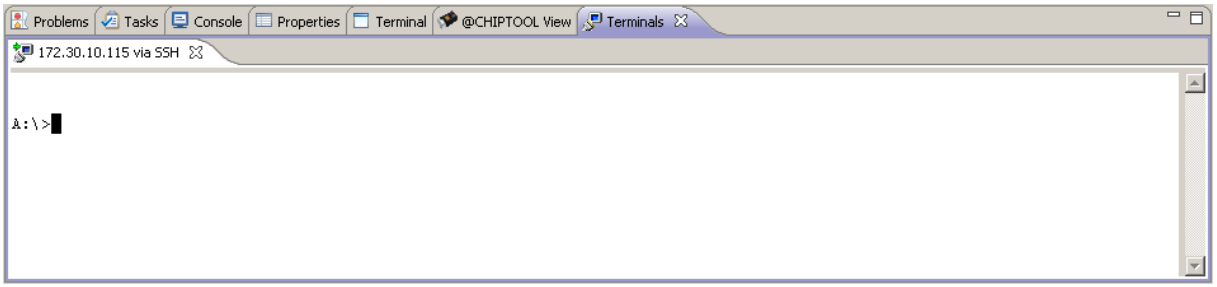

Warning
The authenticity of host '172.30.10.115' can't be established.
RSA key fingerprint is a4:26:0d:9b:ac:5f:71:da:9d:82:3a:60:9a:e8:2b:65.
Are you sure you want to continue connecting?
Yes
No

8. Select *Yes* to create a `known_hosts` file.



9. Press the *OK* button in the next dialog to establish the connection.

A *Terminals* window will show up that lists all established SSH connections. Press Return in the *Terminals* window to see the prompt.



See the next chapter on how to establish the FTP connection to the IPC@CHIP® FTP server.

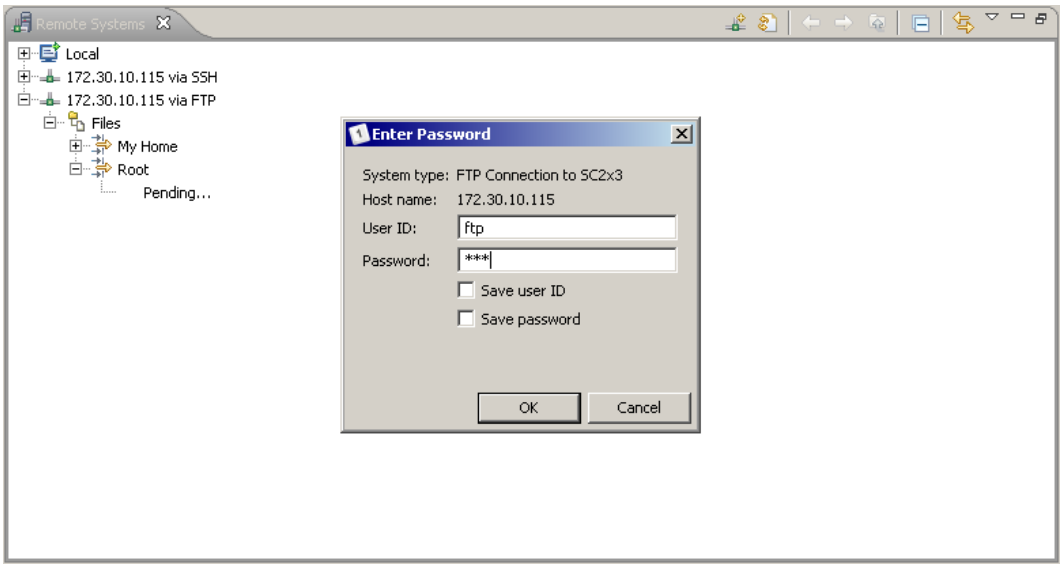
Copyright © 2009 Beck IPC GmbH

2.5 Step 4 – Downloading the DB240 Online demo program

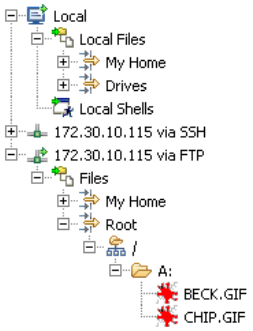
First of all, you should download and extract the `DB240DEMO.ZIP` archive from the Getting Started section of the CD or from the Getting Started webpage to the local hard drive of your PC.

To transfer files to and from your IPC@CHIP® as easy as possible, the ONE-Workbench includes a simple FTP client. This can be used to connect to the @CHIP-RTOS-PPC's FTP server. If you have followed the previous chapter, then you should have already setup a FTP connection in the ONE-Workbench. To open the connection and get file access to your IPC@CHIP®, simply follow these steps:

1. Go to the Remote Systems window and select the *via FTP* tree.
2. Open the *Files/Root* node. A window will appear where you must enter the FTP user name and password.
3. Complete the fields *User ID* and *Password* each with *ftp*



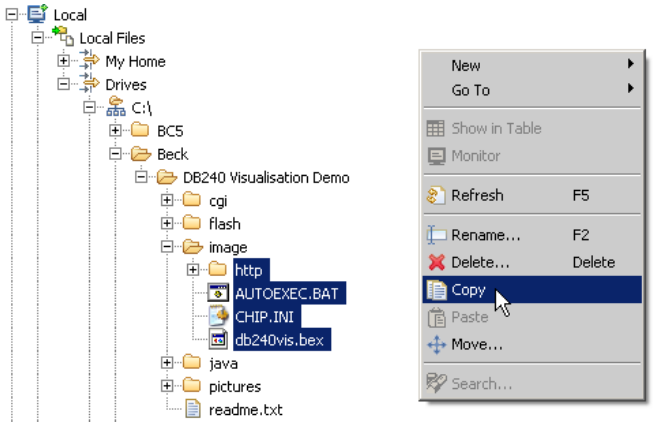
4. Press the *OK* button to establish the connection.
5. Go back to the *Remote Systems* window. If you open the `A:` root directory of the IPC@CHIP®, you now see the content of the IPC@CHIP®'s internal flash drive. The content of the chip's drive may vary depending on how you've configured your IP settings.



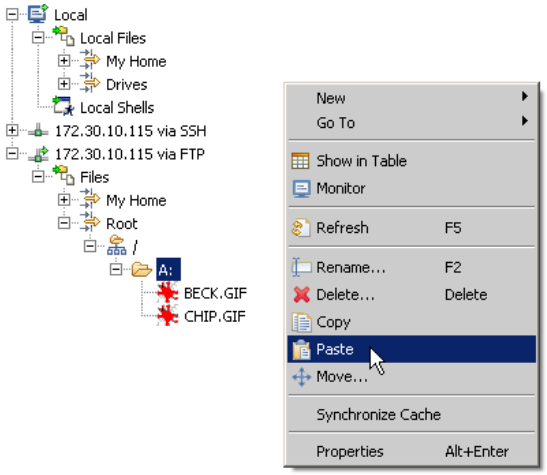
Copyright © 2009 Beck IPC GmbH

You will now upload the DB240 Visualisation Demo that is further described in section 2.6.

6. In the *Local / Local files* tree move to the directory where you have stored the Visualisation Demo.
7. Now change to the subdirectory `Image` of the extracted DB240 Visualisation Demo, which contains all files that have to be deployed on the IPC@CHIP®.
8. Select all files and folders in the `Image` directory.



9. Right-click on one of the files and select *Copy* from the pop-up menu.
10. Go to the IPC@CHIP® root directory `A:`, open the tree so that the files on drive `A:` will show up. Now right-click on `A:` and select *Paste* from the pop-up menu.



The Visualisation Demo will now be copied to the IPC@CHIP®. If the FTP client notifies you that a file named `CHIP.INI` or `AUTOEXEC.BAT` already exists, overwrite it. You will then have to repeat the IP configuration described in section 2.4.2 after rebooting the chip. The file `CHIP.INI` includes information such as the IP settings.

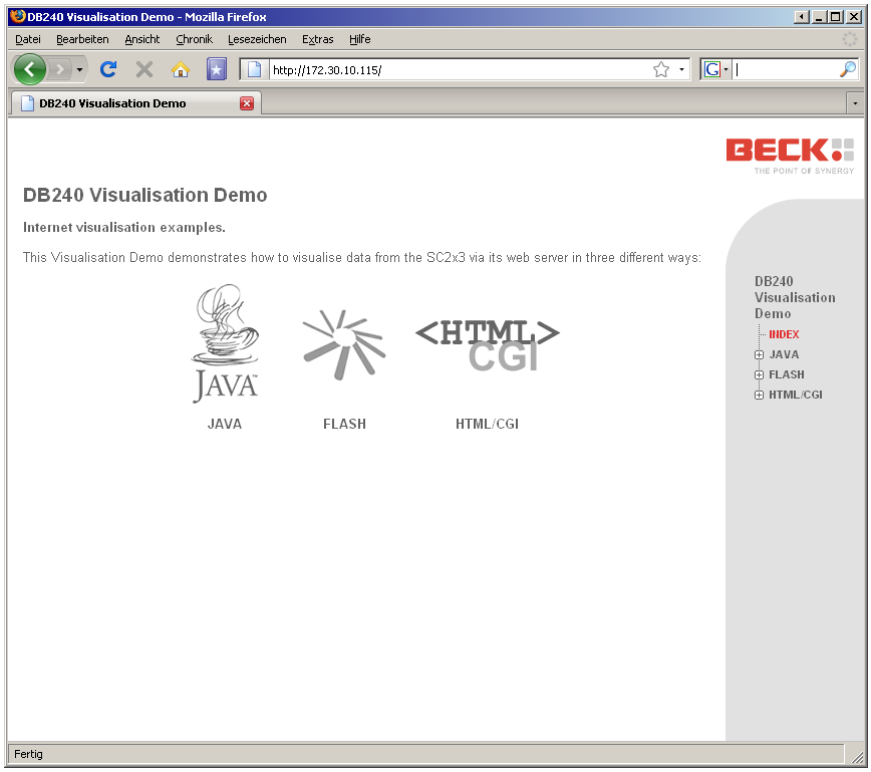
The FTP client provides many more functions than just transferring files. You may also create and remove directories or edit text files situated on the IPC@CHIP® such as the `CHIP.INI`. The handling should be quiet intuitive.

Copyright © 2009 Beck IPC GmbH

2.6 Step 5 – Operating the Demo program

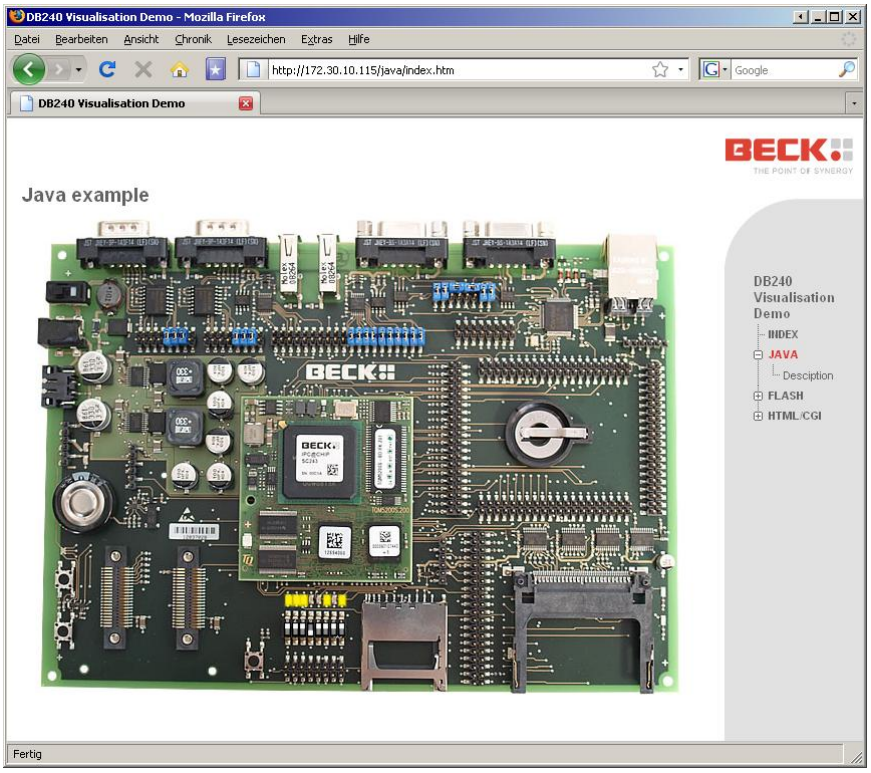
To activate the Visualisation Demo that you have just transferred in section 2.5, reboot the IPC@CHIP® by pressing the DB240's *Reset* button. The Visualisation Demo is now accessible via the IPC@CHIP®'s web server.

Open a web browser like the Internet Explorer or Mozilla Firefox and enter your IPC@CHIP® IP address, like *http://172.30.10.115*. The browser will show the main page of the IPC@CHIP® which now is the Visualisation Demo.



The Visualisation Demo provides three techniques to visualise and modify the digital I/Os of the DB240 – a Java Applet, a Flash Movie and a plain HTML/CGI Page. The Java Applet as well as the Flash Movie shows a picture of the DB240 plate. The I/Os change their colour depending on their state. You may click the output LEDs in order to toggle them.

Copyright © 2009 Beck IPC GmbH



Of course the appropriate browser plug-ins have to be installed for the Java Applet or the Flash Movie to work.

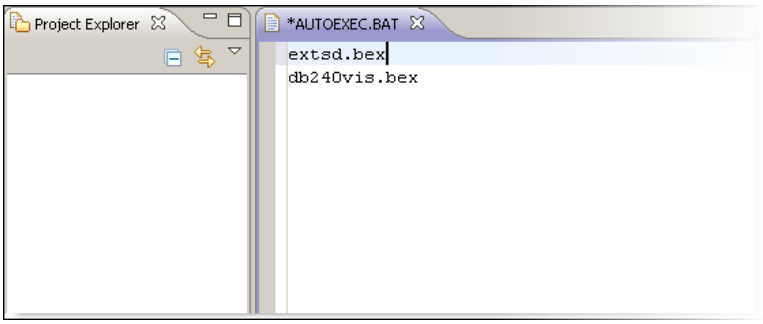
The HTML/CGI Page provides the same functionality but it is not graphically visualised.

Copyright © 2009 Beck IPC GmbH

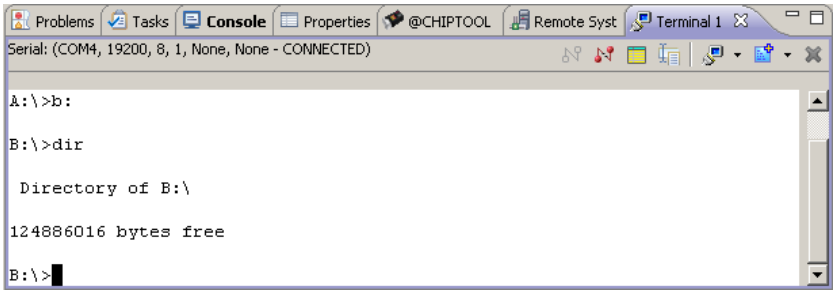
2.7 Step 6 – Configuring SD/MMC-Card

To use the enclosed SD/MMC-Card you'll need additional driver software on the DB240. The driver is available from the Getting Started section of the CD or from the Getting Started website. The driver program needed with SD/MMC-Cards is named `EXTSD.BEX`. Extract this file from the `EXTDISK_Vx.xx.ZIP` archive. The precompiled executable is located in the `EXTSD/image/DB240` folder.

1. Select the *Remote Systems* window and transfer the `EXTSD.BEX` driver file to the root directory of your IPC@CHIP®.
2. Right-click the file `AUTOEXEC.BAT` from the IPC@CHIP®'s flash drive. (If none exists, create an empty file on your local drive and transfer it to the IPC@CHIP®.)
3. Select *Open With / Text Editor* from the pop-up menu.
4. An editor window will open. Add the filename `EXTSD.BEX` in a new line. This will start the driver on boot.



5. Save the file and close the Editor. The file will automatically be reloaded to the chip.
6. Insert the respective SD / MMC card.
7. Reboot the DB240.
8. After rebooting, the IPC@CHIP® will now have a drive B: which represents the content of the card. You can access this drive from the shell or via FTP. Use the command *B:* from the shell to change to the card's file system.



The `EXTSD.BEX` supports several command line options which are needed if you want to mount a certain partition on the card or if you want to use several external disk drives at once. These command line options are described in the source code files of the corresponding driver. These source code files may not be situated on the Getting Started CD. Please check out the download section of our website in this case.

Note that there is also an `EXTUSB.BEX` driver for USB mass storage devices and an `EXTIDE` driver for compact flash cards or hard disks. The `EXTUSB.BEX` driver can also be found on our website. The `EXTIDE` driver is built-in for the SC2x3 as a shell command.

Copyright © 2009 Beck IPC GmbH

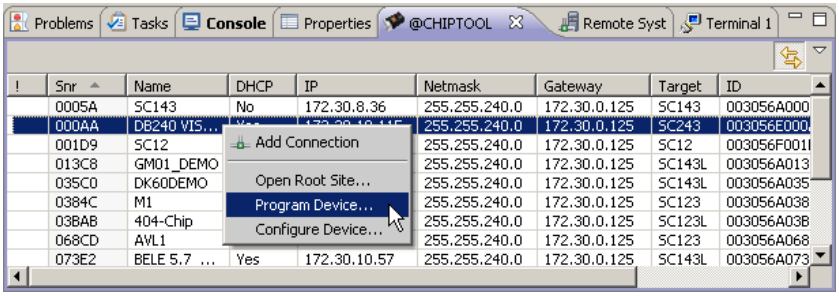
3 Information for advanced users

3.1 Updating the @CHIP-RTOS-PPC

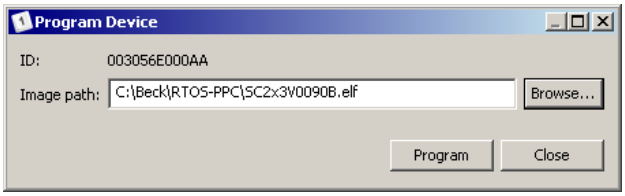
You can download updates for your IPC@CHIP® RTOS-PPC from our website. These updates may then easily be applied using the ONE-Workbench.

See http://www.beck-ipc.com/rtos-ppc

1. Right-click on your IPC@CHIP® in the *@CHIPTOOL View* window.
2. Select *Program Device …* from the pop-up menu.



3. The *Program Device* dialog will appear.
4. Fill the field *Image path* with the path of the RTOS elf file. You may use the button on the right to open a file browsing dialog.



5. Press *Program*. The file will now be uploaded via Ethernet.
6. Wait for the update to complete.

If you are connected to the shell of your IPC@CHIP® while performing an Ethernet update, you will see a lot of 'P' characters while the flash is programmed.

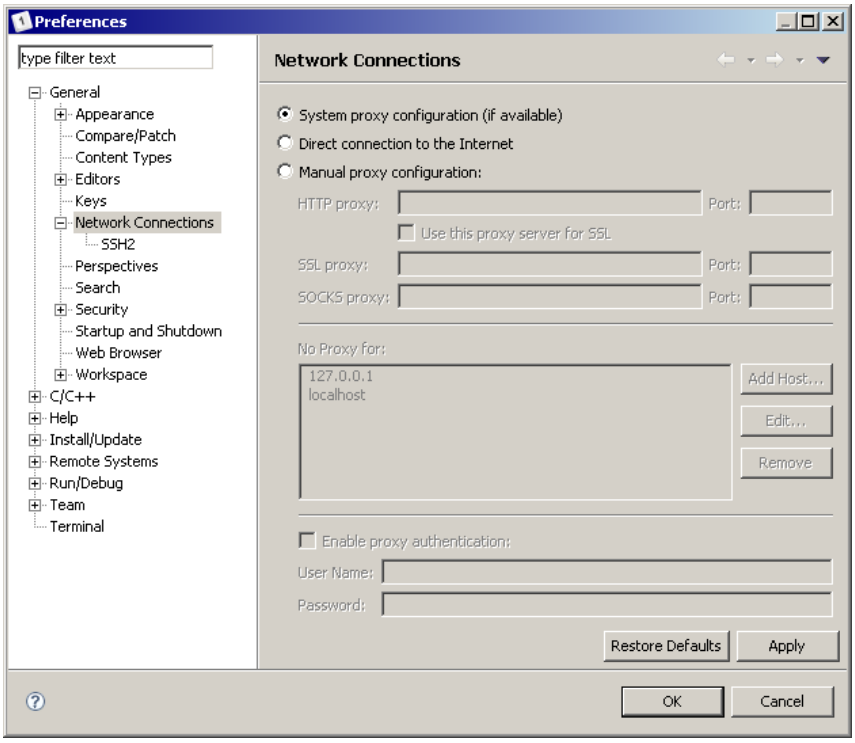
An Ethernet update presumes a running RTOS. So if for some reason your RTOS does not boot any more, you will have to use the serial port update method. This is currently only possible using the @CHIPTOOL application. See the next chapter for more details.

Copyright © 2009 Beck IPC GmbH

3.2 Updating the ONE-Workbench

The ONE-Workbench includes an online update mechanism over the Internet. It can be used to find and install updates and new features.

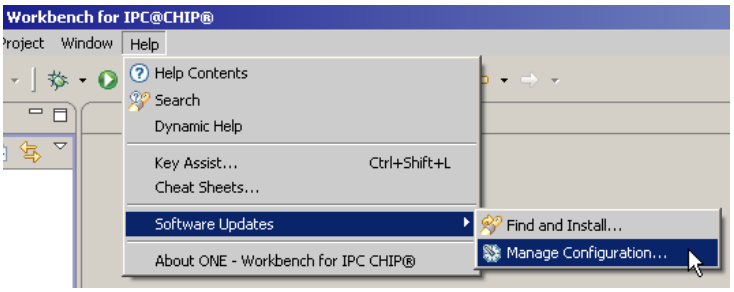
Please first setup your network connection to the Internet in the ONE-Workbench preferences dialog. Choose *Preferences* from the *Window* menu.



Note: If you installed the ONE-Workbench to the Window program files directory ("`C:\Program Files\`"), you will probably have to start the ONE-Workbench with administrator permissions in order to perform an update,

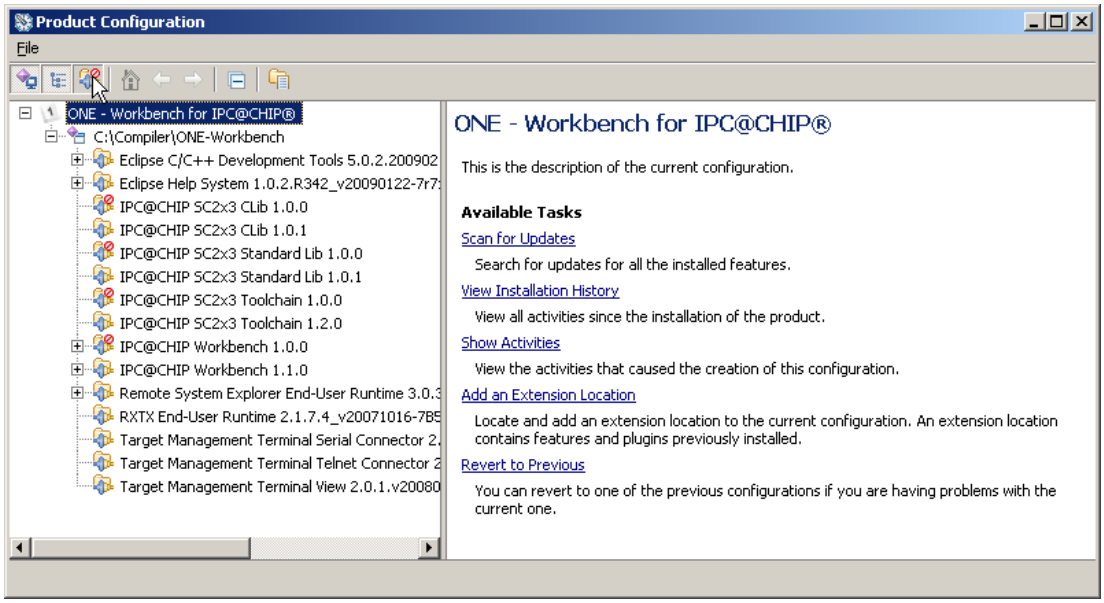
3.2.1 Show currently installed features

Select "*Software Updates/Manage Configuration…"* from the *Help* menu if you want to see the version information of the currently installed features.



In the "*Product Configuration*" dialog you will see all features in the left window tree. The Beck ONE-Workbench extensions to the Eclipse framework are marked with the name "*IPC@CHIP*" in front. If you enable the "*Show Disabled Features*" button in the toolbar of this window, you will also see older versions. This dialog also brings the possibility to revert to an older version.

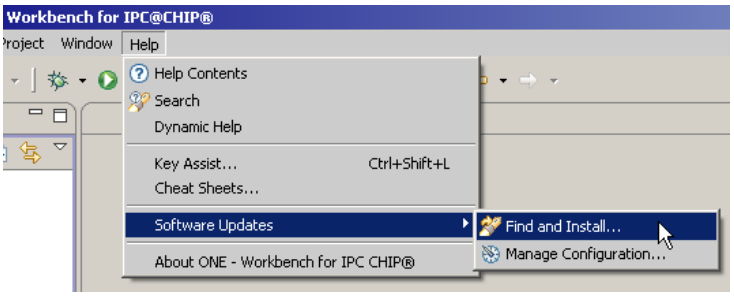
Copyright © 2009 Beck IPC GmbH


Product Configuration
File
ONE - Workbench for IPC@CHIP®
C:\Compiler\ONE-Workbench
Eclipse C/C++ Development Tools 5.0.2.200902
Eclipse Help System 1.0.2.R342_v20090122-7r7
IPC@CHIP SC2x3 CLib 1.0.0
IPC@CHIP SC2x3 CLib 1.0.1
IPC@CHIP SC2x3 Standard Lib 1.0.0
IPC@CHIP SC2x3 Standard Lib 1.0.1
IPC@CHIP SC2x3 Toolchain 1.0.0
IPC@CHIP SC2x3 Toolchain 1.2.0
IPC@CHIP Workbench 1.0.0
IPC@CHIP Workbench 1.1.0
Remote System Explorer End-User Runtime 3.0.3
RXTX End-User Runtime 2.1.7.4_v20071016-7B5
Target Management Terminal Serial Connector 2.
Target Management Terminal Telnet Connector 2
Target Management Terminal View 2.0.1.v20080
ONE - Workbench for IPC@CHIP®
This is the description of the current configuration.
Available Tasks
Scan for Updates
Search for updates for all the installed features.
View Installation History
View all activities since the installation of the product.
Show Activities
View the activities that caused the creation of this configuration.
Add an Extension Location
Locate and add an extension location to the current configuration. An extension location contains features and plugins previously installed.
Revert to Previous
You can revert to one of the previous configurations if you are having problems with the current one.

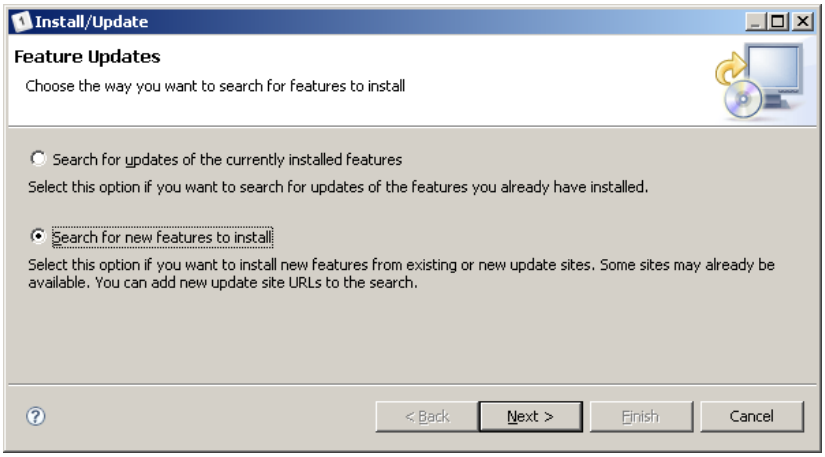
Note: The "*Scan for Updates*" link will not work for major version updates, so please use the update procedure described in the next chapter.

3.2.2 Find and install updates and new features

Select "*Software Updates/Find and Install…"* from the *Help* menu if you want the ONE-Workbench to look for updates and new features.

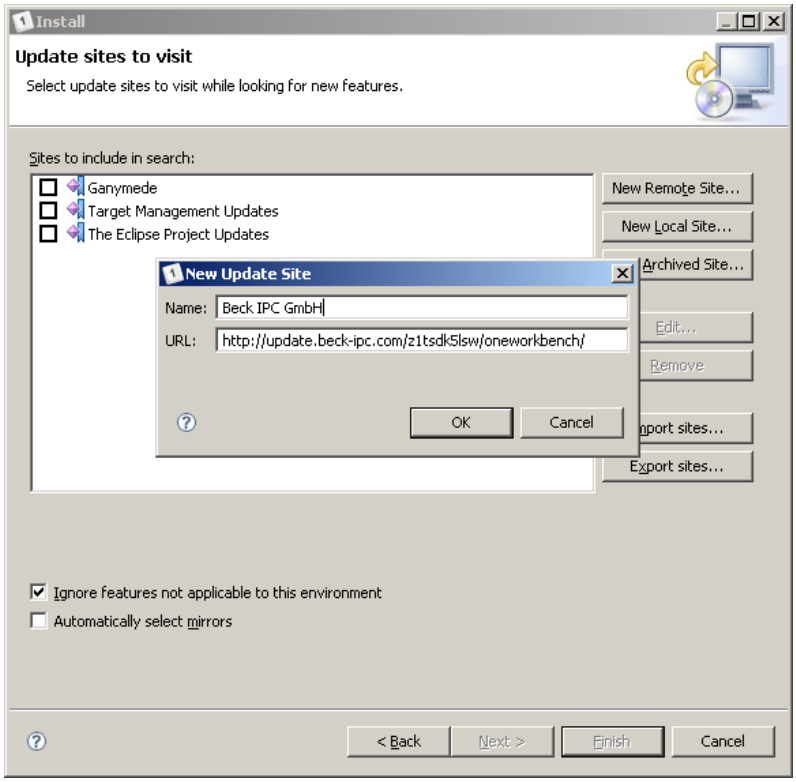


Select "*Search for new features to install*" in the next dialog.


Install/Update
Feature Updates
Choose the way you want to search for features to install
Search for updates of the currently installed features
Select this option if you want to search for updates of the features you already have installed.
Search for new features to install
Select this option if you want to install new features from existing or new update sites. Some sites may already be available. You can add new update site URLs to the search.
< Back
Next >
Finish
Cancel

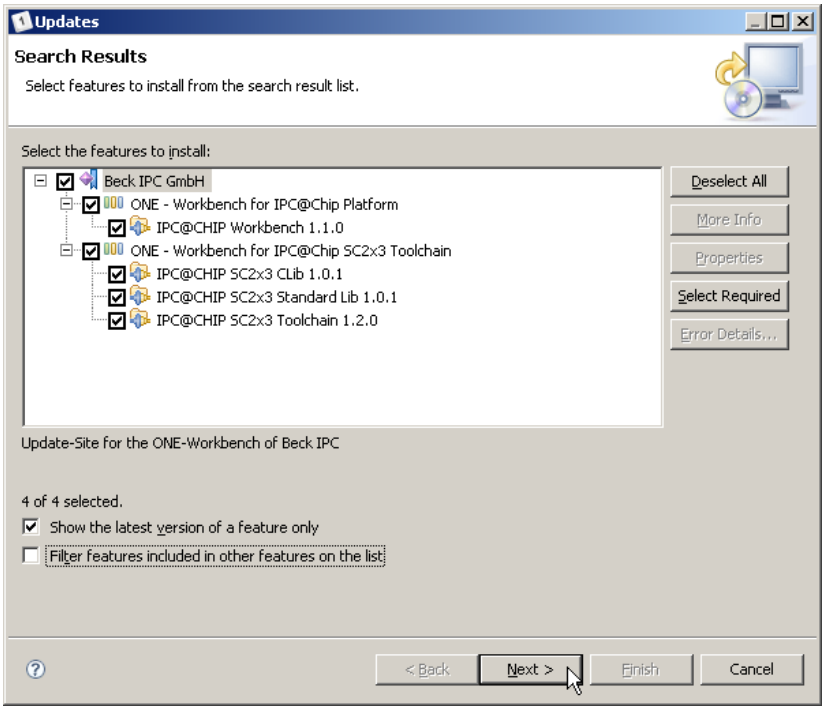
If the "*Beck IPC GmbH*" site is not already included in the provided list of update sites, then please press the "*New Remote Site...*" button and add the URL to the Beck update site.

Copyright © 2009 Beck IPC GmbH



The Name is “Beck IPC GmbH”.
The URL is “http://update.beck-ipc.com/z1tsdk5lsw/oneworkbench/”.

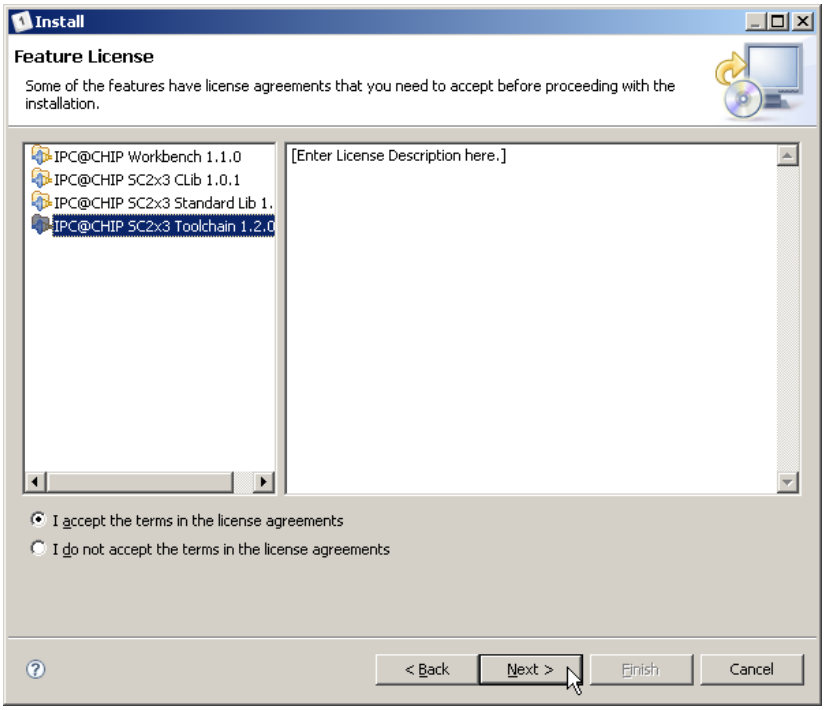
Mark the “*Beck IPC GmbH*” site with a checkmark and click the “*Finish*” button to search for updates and new features.


Updates
Search Results
Select features to install from the search result list.
Select the features to install:
Beck IPC GmbH
ONE - Workbench for IPC@Chip Platform
IPC@CHIP Workbench 1.1.0
ONE - Workbench for IPC@Chip SC2x3 Toolchain
IPC@CHIP SC2x3 CLib 1.0.1
IPC@CHIP SC2x3 Standard Lib 1.0.1
IPC@CHIP SC2x3 Toolchain 1.2.0
Deselect All
More Info
Properties
Select Required
Error Details...
Update-Site for the ONE-Workbench of Beck IPC
4 of 4 selected.
Show the latest version of a feature only
Filter features included in other features on the list
< Back
Next >
Finish
Cancel

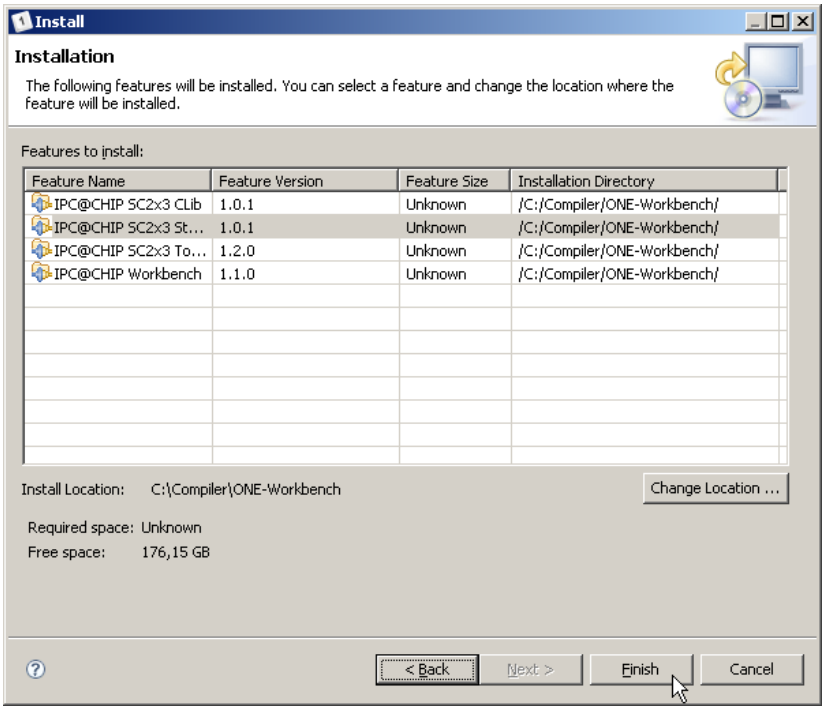
If updates or new features are available, you can select the features that you would like to install. Press the “*Next*” button to continue.

Copyright © 2009 Beck IPC GmbH

In the next dialog you must first accept the terms in the license agreements and then press again the "*Next*" button.


Install
Feature License
Some of the features have license agreements that you need to accept before proceeding with the installation.
IPC@CHIP Workbench 1.1.0
IPC@CHIP SC2x3 CLib 1.0.1
IPC@CHIP SC2x3 Standard Lib 1.
IPC@CHIP SC2x3 Toolchain 1.2.0
[Enter License Description here.]
I accept the terms in the license agreements
I do not accept the terms in the license agreements
< Back
Next >
Finish
Cancel

In the final dialog press the "*Finish*" button to install the updates.


Install
Installation
The following features will be installed. You can select a feature and change the location where the feature will be installed.
Features to install:
Feature Name
Feature Version
Feature Size
Installation Directory
IPC@CHIP SC2x3 CLib 1.0.1 Unknown /C:/Compiler/ONE-Workbench/
IPC@CHIP SC2x3 St... 1.0.1 Unknown /C:/Compiler/ONE-Workbench/
IPC@CHIP SC2x3 To... 1.2.0 Unknown /C:/Compiler/ONE-Workbench/
IPC@CHIP Workbench 1.1.0 Unknown /C:/Compiler/ONE-Workbench/
Install Location: C:\Compiler\ONE-Workbench
Change Location ...
Required space: Unknown
Free space: 176,15 GB
< Back
Next >
Finish
Cancel

After the updates are downloaded, you will be asked to restart the ONE-Workbench.

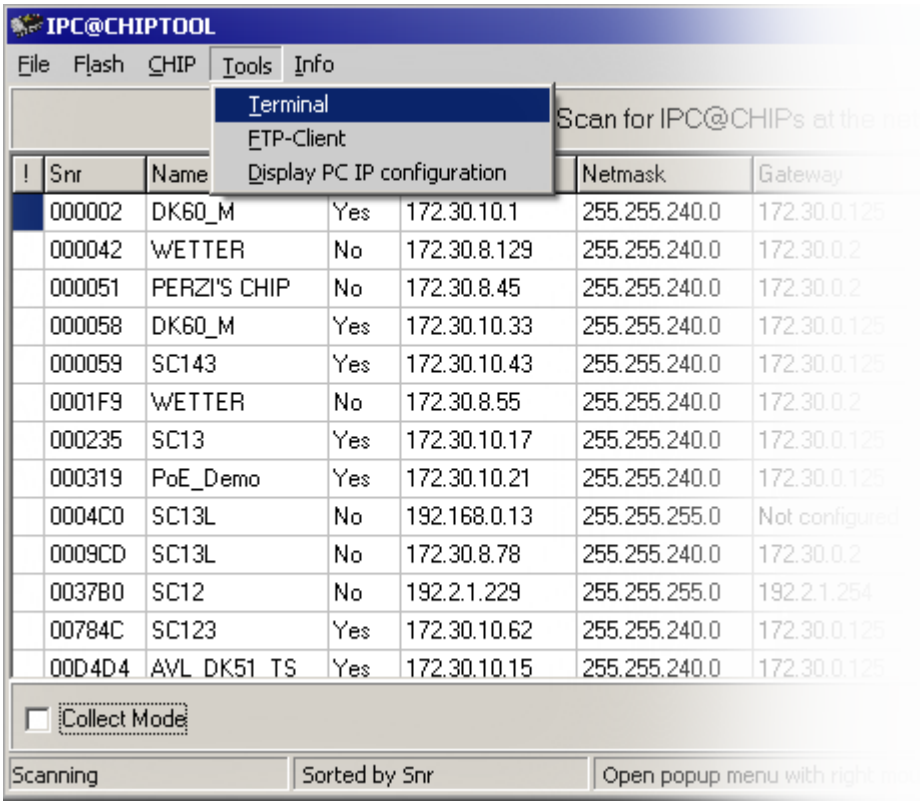
Copyright © 2009 Beck IPC GmbH

3.3 Using the IPC@CHIPTOOL application

The @CHIPTOOL is a communication centre for all IPC@CHIP®-based products, whereas the ONE-Workbench currently only supports the SC2x3 family. It provides functionality for finding IPC@CHIP®s on the network, configuring them, updating their RTOS, transferring files and using their shell. Some features, like a RTOS update over the serial port, are currently only available with the IPC@CHIPTOOL. This program also provides a command-line interface, so it can be used for production.

To install IPC@CHIPTOOL on your PC, execute the Chiptool setup program `CHIPTOOL_INSTALL_Vx.x.x.x.EXE` from the Tools section of the CD or from the Getting Started webpage. After the installation, you are able to start the IPC@CHIPTOOL application. The SC243 requires at least Chiptool version V6.0.3.1.

In the main window you can see all IPC@CHIP®s that could be found on the network. The table displays information such as serial number, name, IP configuration and RTOS version. The main menu provides functionality such as the terminal and an FTP client. The following graphic shows only part of the main window.



The IPC@CHIPTOOL comes with detailed help. Simply press F1 in the main window for further information.

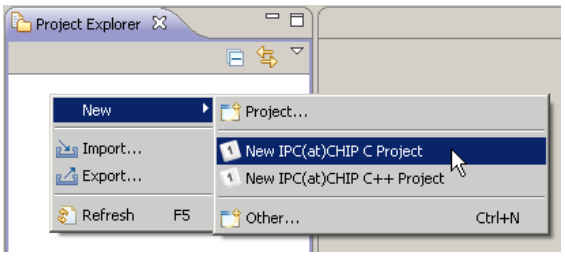
If you're running a firewall, make sure that it does not block any network functions of IPC@CHIPTOOL such as Telnet, FTP or the scanning for IPC@CHIP®s.

Copyright © 2009 Beck IPC GmbH

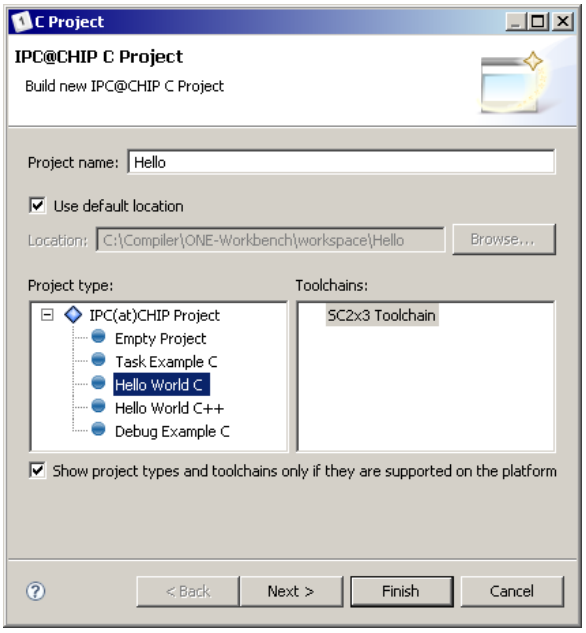
3.4 Developing Software for your IPC@CHIP®

The IPC@CHIP® SC243 features a 32-bit PowerPC processor. The @CHIP-RTOS-PPC uses its own executable format (`*.BEX`). These files are derived from the industry standard ELF files, but they have a relocation table in a condensed format which is specific to the @CHIP-RTOS-PPC. The DK241 Development Kit includes the ONE-Workbench for IPC@CHIP®, which includes an adapted GNU GCC C/C++ toolchain. This compiler produces executables in the above format. You will now implement and deploy your first own program on your DK241 Development Kit. It is presumed that you have installed the ONE-Workbench and followed the above steps.

1. Start the ONE-Workbench. You'll see the main window.
2. Go to the *Project Explorer* window. Right click in the white area and select *New → New IPC(at)CHIP C Project* from the context-menu to create a new project.

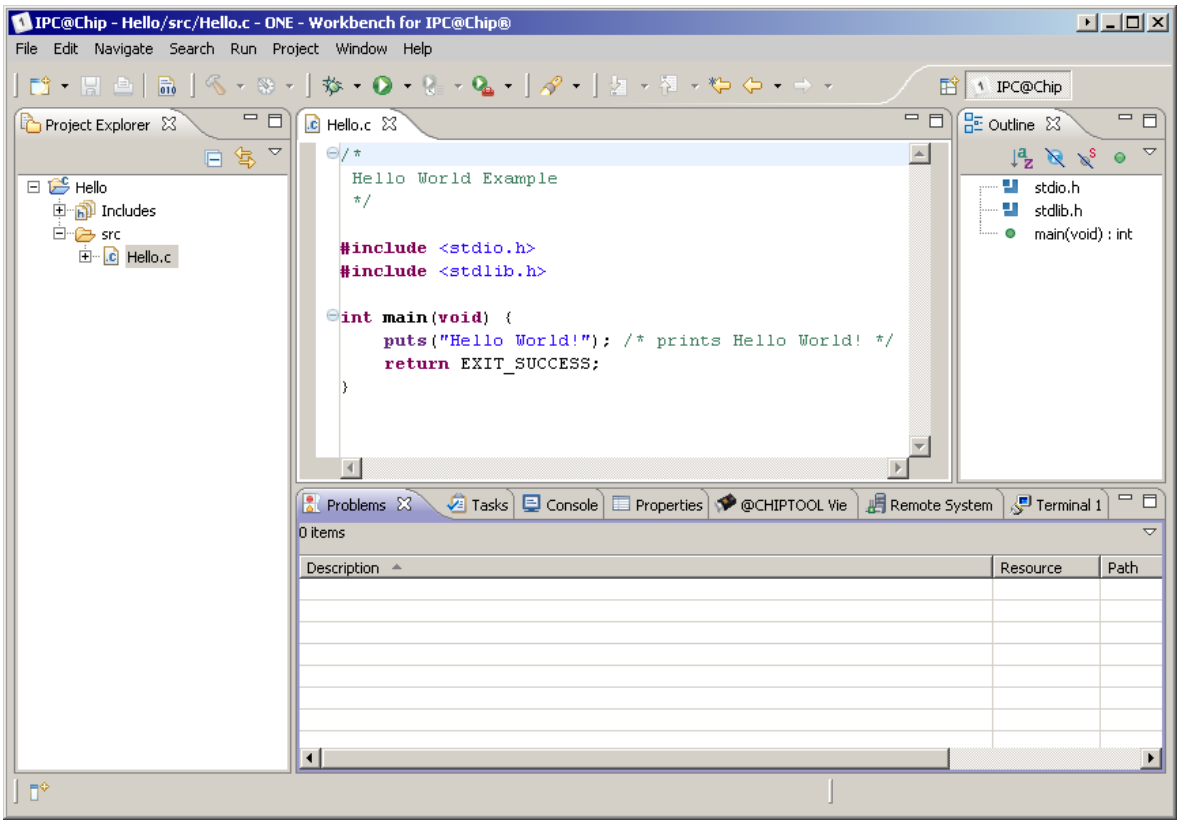


3. Complete the appearing dialog as follows:
   Project name: Enter the name of the project. The default is to create new projects in your workspace folder. If you like to choose a different location, deselect the checkmark in front of *Use default location* and use the *Browse* button to choose the location. For this example we choose the name `Hello`
   Project type: Select the *IPC(at)CHIP Project → Hello World C* option
   Toolchains: Select the *SC2x3 Toolchain*
4. Press the *Finish* button to create the project.



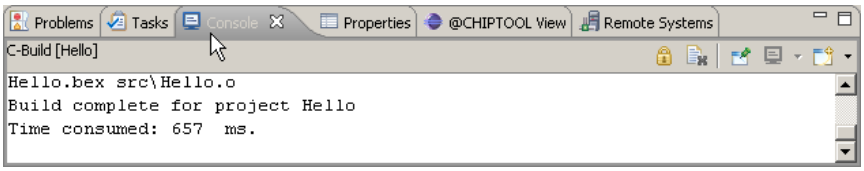
Copyright © 2009 Beck IPC GmbH

5. Now double-click the filename `Hello.c` in the *Project Explorer* view on the left to open the source file.



To get help on a specific function from the IPC@CHIP® C-API simply press ALT-F1. A description of the respective function will open in an internal browser. An overview of the whole API can be found in the main menu of the ONE-Workbench under Help → Help Contents.

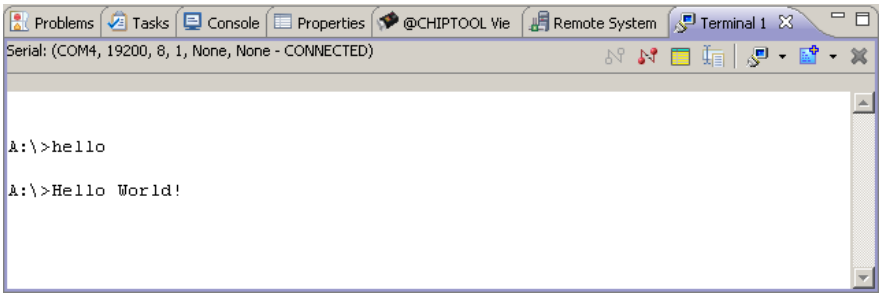
6. Select *Project→ Build Project* from the menu to make the project. The compiler will do its work and show some messages at the *Console* window. A file called `Hello.bex` has now been created in the `debug` directory.



7. Now transfer the `Hello.bex` file to your IPC@CHIP®. Refer back on page 14 to Step 4 – Downloading the DB240 Online demo program, for instructions.

Copyright © 2009 Beck IPC GmbH

8. Open the terminal and type the filename to start the program.



Hint: The last two steps can be automated when using the Run button in the upper toolbar.

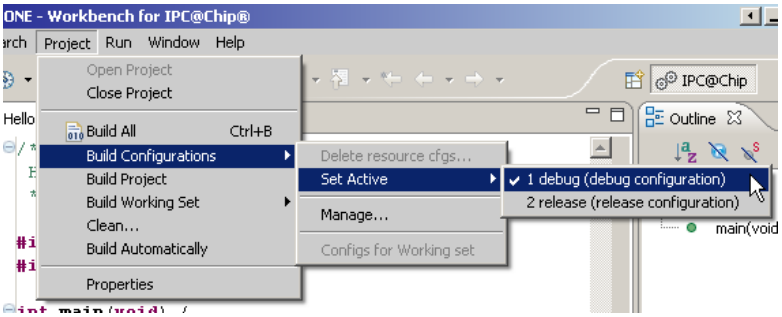
IPC@CHIP Perspective - ONE - Workbench for IPC@CHIP®
File Edit Navigate Search Project Run Window Help
Project Explorer
Run Hello

Copyright © 2009 Beck IPC GmbH

3.5 Debugging your IPC@CHIP® applications

You may debug your IPC@CHIP® applications using the integrated GDB debugger of the ONE-Workbench. It communicates with the IPC@CHIP® built-in server program `PROBE`, which must be running on the IPC@CHIP®. There are CHIP.INI entries to configure the `PROBE` remote debugger. With the factory settings the `PROBE` is started automatically and communicates over the TCP protocol.

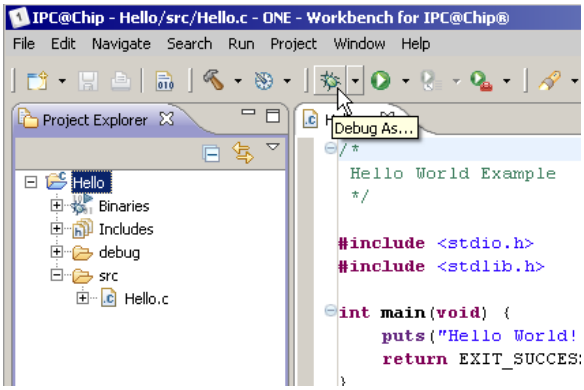
To make your applications debug-able you need to make sure that you have selected the debug configuration for your project.



After (re)compiling the project, an `ELF` file will be created which includes all information that the debugger needs for source-level debugging. This file is only needed on the PC side. Make sure that the Project Explorer lists the *Binaries* node. If the *Binaries* node is not shown select the project node and press F5 for a refresh.

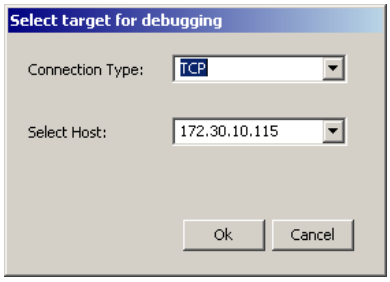
You are now ready to debug your program.

1. Click on the green bug icon in the toolbar to debug your program.

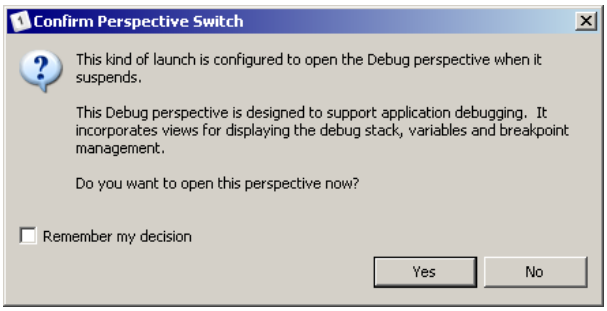


Copyright © 2009 Beck IPC GmbH

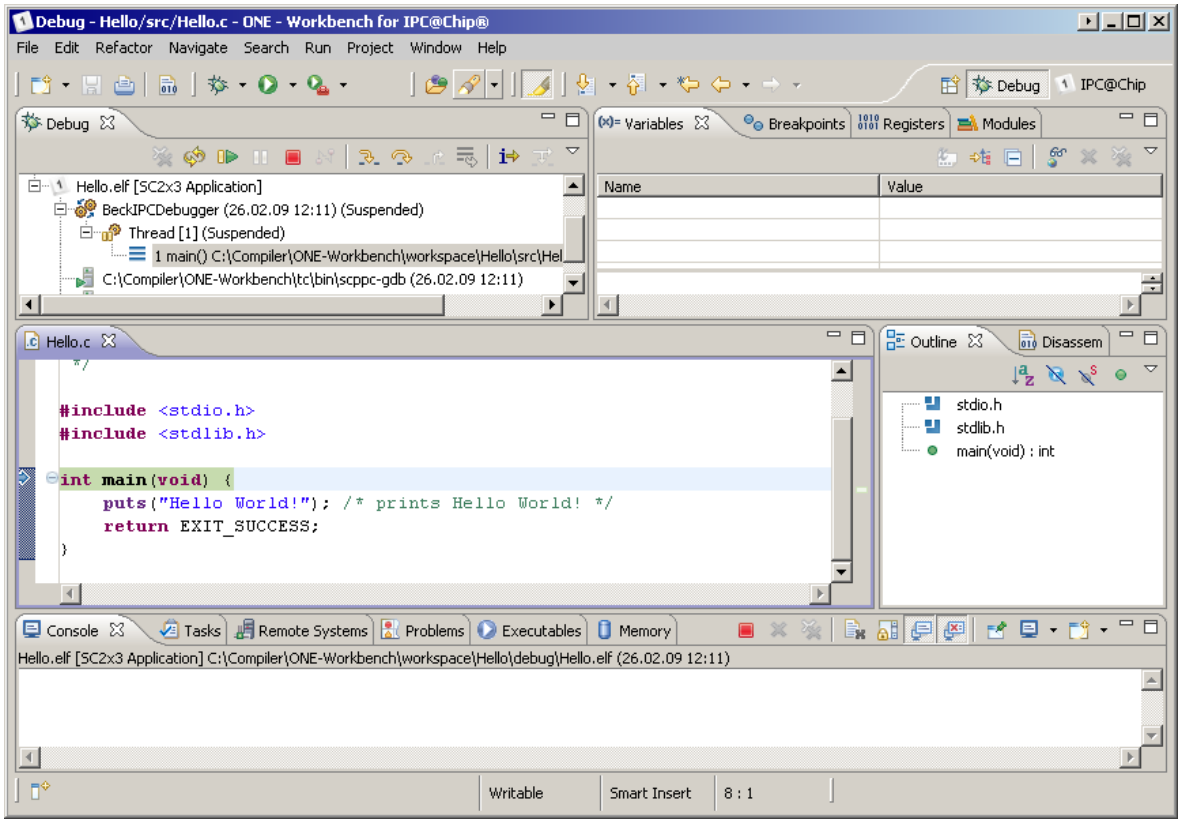
2. Select *TCP* as *Connection Type* and the IP of your IPC@CHIP®.



3. You will be asked if the Debug perspective should be opened. Select *Yes*.

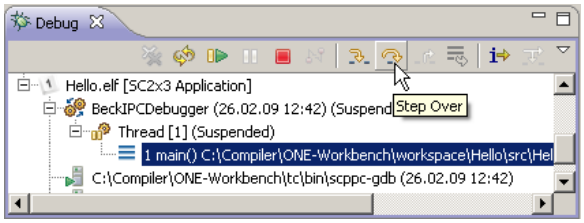

Confirm Perspective Switch
This kind of launch is configured to open the Debug perspective when it suspends.
This Debug perspective is designed to support application debugging. It incorporates views for displaying the debug stack, variables and breakpoint management.
Do you want to open this perspective now?
Remember my decision
Yes
No

4. The Debug perspective will now open and the program is stopped at the start of the main() function.



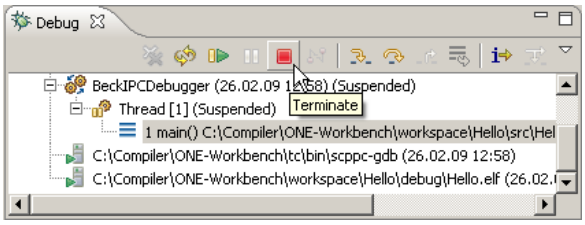
5. You can now control the execution of your program. You can select *Resume*, *Step Into* or *Step Over* from the debug toolbar. Each time you step, one source-level instruction of your program will be executed and the instruction pointer (blue arrow) will move on to the next instruction.

Copyright © 2009 Beck IPC GmbH



6. Or you can set Breakpoints. To set a breakpoint click in the source file window on the grey area in the left-most position. A breakpoint symbol will appear.

7. When you're done, select the *Terminate* icon from the toolbar.



The debugger provides much more functionality, but this goes beyond the scope of this Getting Started document.

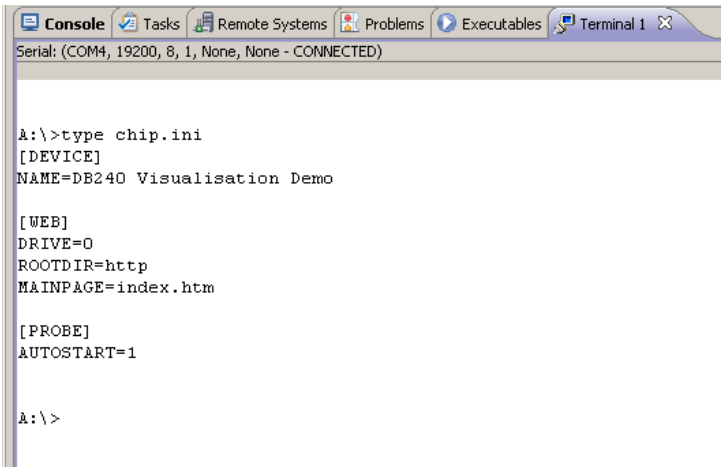
Copyright © 2009 Beck IPC GmbH

3.6 Miscellaneous

This chapter contains miscellaneous information on the @CHIP-RTOS-PPC.

3.6.1 The CHIP.INI configuration file

The @CHIP-RTOS-PPC holds its non-volatile configuration information in a file named `CHIP.INI` found in the root directory of the flash disk drive. This ASCII text file can be adjusted either directly with the `EDIT` command over the shell or over the FTP client of the ONE-Workbench. The figure below shows a few example entries that might be found in a `CHIP.INI` file.



Refer to the @CHIP-RTOS-PPC API Documentation for a complete specification of the entries possible in this configuration file.

3.6.2 The command shell

The @CHIP-RTOS-PPC has a command shell task which accepts commands from a standard I/O device. This standard I/O device can be UART1, UART3, the Telnet port and, if the respective public keys are installed, the SSH port. The selection is controllable from the `CHIP.INI` file.

The standard I/O can be configured to be shared between application programs and the @CHIP-RTOS-PPC command shell. A hot key (by default Ctrl + F) will switch the standard I/O focus between application and RTOS command shell.

The HELP command gives an overview of all commands. Refer to the IPC@CHIP® API Documentation for a detailed description of the shell commands.

3.6.3 The web server

The @CHIP-RTOS-PPC includes a web server via which you can access HTML pages situated on your IPC@CHIP®'s flash with your web browser. These HTML pages may include multi-media content such as images, Java Applets, Flash movies and more. The web server also includes a CGI and SSI interface. This allows the creation of pages with dynamic content in your applications. To achieve this, a certain function is registered with the web server that will be called if a certain web address is requested. This function will then build the page.

By default there are two active CGI pages on your IPC@CHIP®. One of them, the main page, is disabled when uploading the DB240 Visualisation Demo. The second one is still accessible under the address http://172.30.10.50/ChipCfg, assuming that 172.30.10.50 is your

Copyright © 2009 Beck IPC GmbH

IPC@CHIP®'s IP address. This page shows some more or less useful information on the chip's configuration.

When creating your own web pages for your IPC@CHIP®, you canuse the DB240 Visualisation Demo as an example.

@CHIP-RTOS-PPC also features a SSL (Secure Socket Layer) web server. SSL uses cryptographic methods to establish a trusted connection between the SSL web server and the browser, which makes it impossible for third parties to spy on or manipulate transferred data. To set up the SSL web server you need a certificate which identifies your IPC@CHIP® unambiguously to the user. You may create such a certificate yourself or buy one from a so-called Certification Agency. This agency then signs the certificate and guarantees its authenticity. To register the certificate with the web server and enable the SSL functionality, you have to use the `CHIP.INI` configuration file. Besides the certificate which consists of two files, you need to generate a file containing private and public keys used by the cryptographic functions.

3.6.4 The PPP server/client

PPP (the Point to Point Protocol) is a mechanism for running TCP/IP over a serial link - be that a direct serial connection (using a null-modem cable), or a link made using an analogue modem.

Other computers can dial into the IPC@CHIP® PPP server and communicate via the TCP/IP link using FTP, Telnet, HTTP, etc. in the same manner as with an Ethernet TCP/IP link. One major difference between a PPP and an Ethernet connection is of course the speed. The IPC@CHIP®'s Ethernet interfaces operate at 100 mbps, whereas an analogue modem operates at speeds up to 56 kbps.

PPP is strictly a peer to peer protocol. There is no difference between the machine that dials in and the machine that is dialled into. However, it is still useful to think in terms of servers and clients. When you dial into a site to establish a PPP connection, you are considered the client. The machine to which you connect (e.g. the IPC@CHIP®) is considered the server. The @CHIP-RTOS-PPC operating system provides TCP/IP API functions for either side of this client/server relationship.

PPP at the IPC@CHIP® includes the sub protocols LCP and NCP (IPCP). Supported authentication protocols are PAP and CHAP.

Copyright © 2009 Beck IPC GmbH

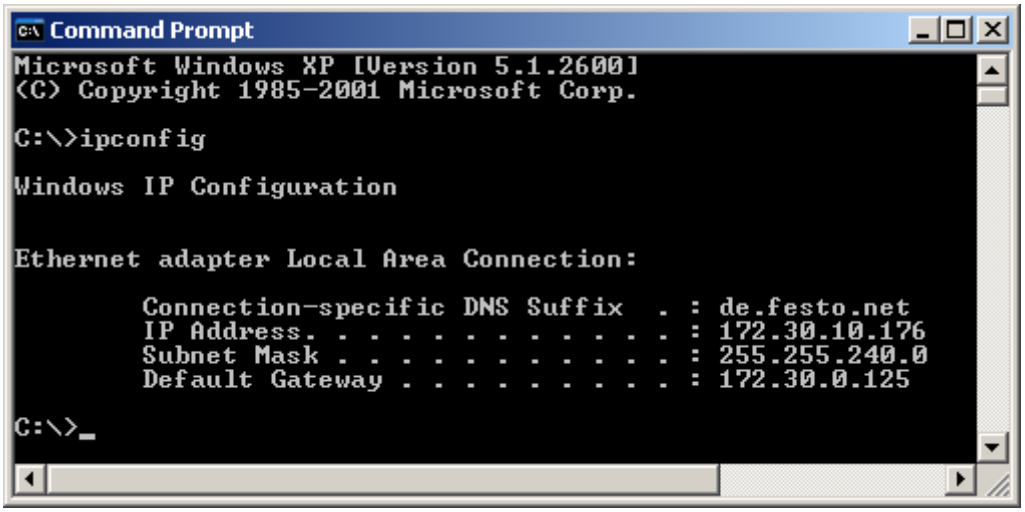
3.6.5 IPv4 Basics

On an IP network IP addresses are used to define the source and the destination of a data packet. An advantage of IP addresses is that they also hold information on a device's physical location in a large network. This is best explained by an example:
Suppose you have the address 172.30.10.50 and the network mask 255.255.240.0. Let's look at this data in binary format:

IP address	172.30.10.50	10101100.00011110.00001010.00110010
Network mask	255.255.240.0	11111111.11111111.11110000.00000000
Logical AND of both		10101100.00011110.00000000.00000000

The first 20 bits of the network mask are 1. This means that the first 20 bits of the IP address identify the network. The last 12 bits (those that are 0 in the network mask) identify the device within this network. This means that in the given example we can give the devices in our network the addresses 172.30.0.1 through 172.30.15.254. The lowest (here 172.30.0.0) and the highest address (here 172.30.15.255) of a network are reserved and cannot be assigned to a host. The lowest address identifies the network, the highest one is a broadcast address, used for a broadcast to all devices on this section of the network. Messages sent to this address are received by all devices.

Any address that is not within the range of the current network is passed to the gateway address (the router). A gateway is a physical machine, hardwired into two or more physical networks. It knows how to transfer an IP packet from one network to another.
You can find the network configuration of the network that you are connected to by entering the command `IPCONFIG` in a Windows command shell.


Command Prompt
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.
C:\>ipconfig
Windows IP Configuration
Ethernet adapter Local Area Connection:
Connection-specific DNS Suffix . : de.festo.net
IP Address. . . . . : 172.30.10.176
Subnet Mask . . . . . : 255.255.240.0
Default Gateway . . . . . : 172.30.0.125
C:\>

On a network, no two devices are allowed to have the same address.

Copyright © 2009 Beck IPC GmbH

3.6.6 IPv4/v6 Dual Stack

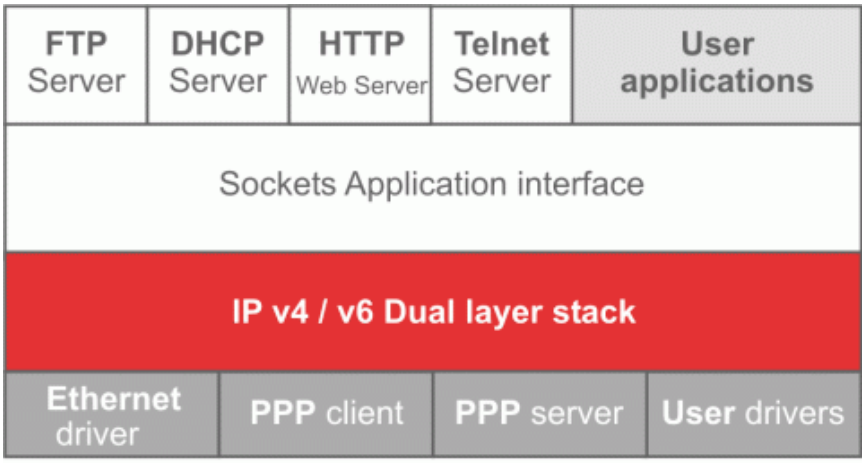
In addition to the IPv4 protocol, the TCP/IP protocol stack of the IPC@CHIP® SC2x3 provides IPv6.

IPv6 is sometimes called the Next Generation Internet Protocol, or IPng. The current Internet and most of corporate or private intranets use IPv4, developed in 1970s and 1980s. Now with IPv6, the first significant upgrade of the Internet protocol has been started. The compelling reason behind the formation of IPv6 was lack of address space, especially in the heavily populated countries of Asia such as China.

The most important changes from IPv4 to IPv6 can be summarized as follows:

- Expanded addressing and auto configuration mechanisms
- The address size for IPv6 is 128 bits, instead of 32 bits for IPv4. This solves the problem of limited address space and offers much simpler address configuration.

The TCP/IP stack of the IPC@CHIP® works as a dual layer stack. If the IPv6 protocol is enabled on the IPC@CHIP®, all TCP/IP applications (e.g. HTTP, Telnet or FTP servers) are accessible via both IPv4 and IPv6. The provided socket interface supports both protocol types.



For more information on IPv6 on the IPC@CHIP® please refer to the IPv6 section of the @CHIP-RTOS-PPC API documentation.

Copyright © 2009 Beck IPC GmbH

4 Appendix - Where to get further Information

4.1 Getting Started

All software referred to in this document can be downloaded from out Getting Started website at http://www.beck-ipc.com/gettingstarted.

4.2 Updates, tools and examples

Regularly check out IPC@CHIP® website at http://www.beck-ipc.com/en/download for:

- @CHIP-RTOS-PPC updates
- New APIs
- Example programs that you may use as a base for your own applications
- Useful tools for your development
- Application notes

Regularly check for ONE-Workbench online updates, see chapter 3.2

4.3 Support

If you need support please…

1. Check our support website at http://www.beck-ipc.com/en/support
2. Visit our forum at http://forum.beck-ipc.com/. Here you will find a lot of answers to customer problems and you can also add your own ideas and questions.
3. Contact our support at support@beck-ipc.com. Of course you may also contact us by phone. But please use email if possible. This not only reduces the time you spent on the phone, it also allows us to better manage the questions and identify problem areas.

4.4 Order numbers

Most parts of the Development Kit DK241 can be ordered separately. Please visit our online shop. These are the order numbers.

DK241 Development Kit	563965
DB240 Development Board	563964
ZK14 serial null-modem cable	195837
PSE10 power supply	538934
ONE-Workbench for IPC@CHIP®	566784
SC243	563963

Copyright © 2009 Beck IPC GmbH

Copyright © 2009 BECK IPC GmbH All rights reserved
No part of this document may be copied or reproduced in any form or by any means without the prior written consent of BECK IPC GmbH. The information in this document is subject to change without notice. Devices sold by BECK IPC GmbH are covered by warranty and patent indemnification provisions appearing in BECK IPC GmbH Terms and Conditions of Sale only.
BECK IPC GmbH MAKES NO WARRANTY, EXPRESS, STATUTORY, IMPLIED OR BY DESCRIPTION, REGARDING THE INFORMATION SET FORTH HEREIN OR REGARDING THE FREEDOM OF THE DESCRIBED DEVICES FROM INTELLECTUAL PROPERTY INFRINGEMENT. BECK IPC GmbH MAKES NO WARRANTY OF MERCHANTABILITY OR FITNESS FOR ANY PURPOSE.
BECK IPC GmbH shall not be responsible for any errors that may appear in this document. BECK IPC GmbH makes no commitment to update or keep current the information contained in this document.

Life critical applications - BECK products are not authorized for use as critical components in life support devices or systems unless a specific written agreement pertaining to such intended use is executed between the customer and BECK prior to use. Life support devices or systems are those which are intended for surgical implantation into the body, or which sustain life, and whose failure to perform, when properly used in accordance with instructions for use provided in the labeling, can be reasonably expected to result in significant injury to the user. BECK IPC GmbH customers using or selling these products for use in such applications do so at their own risk and agree to fully indemnify BECK IPC GmbH for any damages resulting from such application.
Right to make changes - Beck IPC GmbH reserves the right to make changes without notice in the products, including software, described or contained herein in order to improve design and/or performance. Beck IPC GmbH assumes no responsibility or liability for the use of any of these products.